EDS INDIANA TITLE XIX

Autosystem Overview

Turnover Library Reference Number: SYAP10006

Document Management System Reference: Autosystem Overview SYAP10006

Address any comments concerning the contents of this manual to:

Revision History

Document Version Number	Revision Date	Revision Page Number(s)	Reason for Revisions	Revisions Completed By
Version 1.0	August 2000	All	Format and Content Updates	Jeff Strand (SME), Karen Girgis

Table of Contents

Section 1: Introduction

Document Conventions

This document uses specific typographic conventions. These standard text formats and visual clues make a document easier to use and understand.

Table 1.1 – Document Conventions

Style	Use
bold	Individual characters, emphasized words, text the user is required to type, or computer menu item selections are printed in boldface type.
italics	Words defined in text, titles, references, rules and laws, foreign words, mathematical variables in equations or text, directory paths and file names are printed in italics.
`Courier New`	Samples of text appearing on a computer monitor as computer source code are printed in Courier New. This typeface resembles the print produced on a typewriter; each letter occupies the same amount of space on the page.
Notes: Helpful hint.	Helpful hints and supplementary information are indented, italicized, and boxed.

Section 2: AutoSys

Overview

AutoSys is the software used by Indiana Title XIX to schedule and execute the batch cycle based on user-defined dependencies and time parameters. Whenever a job runs under AutoSys, the status of the job and its completion code are stored in the database for future reference.

In AutoSys, a BOX is something that has one or more jobs or boxes grouped under it. There can be a BOX within a BOX, or there can be a JOB within a BOX. There cannot be a JOB within a JOB.

> *Note: The following AutoSys naming convention tables only represent a subset of possible systems. This is not an exhaustive list of possibilities.*

Following are the *AUTOSYS BOX* naming conventions.

Table 2.1 – *AUTOSYS BOX* Naming Conventions ESSF###

Naming Convention	Description
E is the environment.	
P	Production
M	Model Office
SS is a two-character representation of the subsystem.	
CL	Claims
TP	Third Party Liability
EL	Eligibility
PA	Prior Authorization
FI	Financial
PR	Provider
MA	MARS
SR	SURS, etc.
F is the frequency.	
D	Daily
W	Weekly
M	Monthly

(Continued)

Table 2.1 – *AUTOSYS BOX* Naming Conventions ESSF###

Naming Convention	Description
A	Annually
R	On Request
O	Other
### is usually a unique sequence number.	

Table 2.2 – *AUTOSYS JOB* Naming Conventions SSSEF###

Naming Convention	Description
SSS is a three character representation of the subsystem.	
CLM	Claims
TPL	Third Party Liability
ELG	Eligibility
PAU	Prior Authorization
FIN	Financial
PRV	Provider
MAR	MARS
SRG	SURS, etc.
E is the environment.	
P	Production
M	Model Office
F is the frequency.	
D	Daily
W	Weekly
M	Monthly
A	Annually
R	On Request
O	Other
### is usually a unique sequence number.	

> *Note: The naming standards for the ACTUAL job (the job script) are the same as the AutoSys standards, with the only exception being that the environment character is a **J** in the ACTUAL job name.*

Usually, the AutoSys job name will be the same as the actual job name, with only the environment character being different, but there are some exceptions.

For example, there are some cases when the same job needs to run in two separate boxes. For example, ELGJD007 runs both in the PELDICES box and in PELMICES box. In this situation, the AutoSys job could not be named the same for both jobs; they must have unique names; for example, ELGPD007 and ELGPM107. The actual UNIX job being run can only be determined by typing **jr <JOBNAME> -q.**

Understanding Jobs

There are three job types in AutoSys. These types are described in the following table:

Table 2.3 − AutoSys Job Types

Job Type	Description
Command	Executes a job script. IndianaAIM uses this type to actually run a job.
Box Job	A container of other jobs—a box is used to group jobs that go together. For example, all the daily jobs for the recipient subsystem may be placed in one box. ***Best Practice Note: Place jobs into box in the order they are to be run. AutoSys Note:** If a job is added after a box has already been built, it will be added to the bottom of the jil unless the whole box and all jobs are deleted and re-entered.*
File Watcher	Monitors the creation of a specified file in a specified directory on a specified machine

Viewing the Schedule and/or Dependencies of an AutoSys Job/Box

To view the schedule and/or dependencies of an AutoSys job, type the following: **jr <job/box name> -q**. For example, if you type the following: **jr CLMPD795 -q**.

The result will appear as shown in the following figure.

```
/* ----------------- CLMPD795 ----------------- */

insert_job: CLMPD795    job_type: c
box_name: PCLD700
command: autojob.sh CLMJD795
machine: inprodonline
#owner: dsibprod@dsibsun0
permission: mx,me
condition: SUCCESS(PFIWKLY) AND SUCCESS(PCLDRCYC_UPDT) AND
SUCCESS(dbap_bounce_inaimp1)
description: "Deny CCF's older than 45 days"
alarm_if_fail: 1
```

Figure 2.1 − View of Schedule

The **command** line tells what ACTUAL job script is executed by this AUTOSYS job.

The **machine** line tells on what machine the job runs.

The **#owner** line tells what identification someone must be in order to do anything to that job. For example, to force start this job, the user mus be **dsibprod** (type **su dsibprod**).

> *Note: The machine listed on this line is not the machine the job runs on, simply the one the original jil was set up on.*

The **description** line gives a very brief description of the job.

Section 3: Common Job Definition Field Descriptions

Table 3.1 − AutoSys Label Cross Reference

Label	Description
insert_job:	Creates a new job
delete_job:	Deletes a job from the database
update_job:	Updates a job currently on the database
box_name:	Name of the box this job is in
date_conditions:	Indicates if there are date conditions (1=yes, 0=no)
days_of_week:	Days job/box will run
run_calendar:	AutoSys calendar days that job will run
start_times:	Starting times for job/box
command:	UNIX command, shell script, application program that is to be run
machine:	Machine that this job will run on
#owner:	The owner of the job

permission:	Controls access to job definitions and execution permissions allowed
	
	    gx: group execute
	
	    ge: group edit
	
	    mx: authorized user can execute on any machine
	
	    me: authorized user can edit on any machine
	
	    wx: world execute
	
	    we: world edit

condition:	Conditions to be met before a job can start
description:	Description of the job
alarm_if_fail:	Indicates whether an alarm should be posted to the event processor

Section 4: AutoSys Databases

Table 4.1 − Sun Machines on which the Databases Run

Environment	Database	Machine
Model Office	INAIMM1	DSIBSUN0
Production	INAIMP1	DSIBSUN1 & DSIBSUN2 & DSIBSUN3
AutoSys		DSIBSUN3

Table 4.2 − AutoSys Machine Alias'

Machine	Alias'
dsibsun0	inmodce01, inmodhist, inmodonline, inmodsur, inmodmar
dsibsun1	inprodce01
dsibsun2	Inprodhist, inprodonline
dsibsun3	Inprodsur, inprodmar

```
/* ---------------- PCLD700 ---------------- */

insert_job: PCLD700   job_type: b
#owner: dsibprod@dsibsun3
permission: mx,me
date_conditions: 1
days_of_week: mo,tu,we,th
exclude_calendar: holiday
start_times: "19:00"
condition: SUCCESS(dbap_bounce_inaimp1)
description: "This box extracts data correction requests from the on-line databa
se"
alarm_if_fail: 1
```

Figure 4.1 − Example of AutoSys Job Format

Boxes Include the Following Information

- Date conditions
- Days of week
- Include/exclude calendars
- Start times
- Job type B

```
      /* ---------------- ELGPD010 ----------------- */


insert_job: ELGPD010    job_type: c
box_name: PELDICES
command: autojob.sh ELGJD010
machine: inprodmar
#owner: dsibprod@dsibsun2
permission: mx,me
condition: SUCCESS(ELGPD003) AND SUCCESS(ELGPD005) AND SUCCESS(ELGP
D007) AND NOTRUNNING(ELGPM050) AND SUCCESS(ELGPW010) AND SUCCES
S(ELGPW020) AND NOTRUNNING(MGDPD100-2) AND SUCCESS(PREIXTRT) AND
SUCCESS(CLMPD100) AND NOTRUNNING(MGDPD600)
description: "ICES Recipient Eligibility Update"
alarm_if_fail: 1
```

Figure 4.2 – Example of Autosys Box Format

Jobs Include the Following Information

- Box names
- Command lines
- Machines
- Job type C
- Condition lines wrap

```
 /* ---------------- PELWATCH ----------------- */


insert_job: PELWATCH    job_type: f
box_name: PELDICES
machine: inprodmar
#owner: dsibprod@dsibsun2
permission: mx,me
date_conditions: 1
run_calendar: DAILYICES
start_times: "05:05"
description: "ICES Eligibility File Watcher - Production"
watch_file: /export/customer/dsib/prod/data/eld7001.dat
watch_interval: 60
alarm_if_fail: 1
```

Figure 4.3 – Example of an AutoSys Watcher Job Format

Creating and Updating Jils

To look at an existing JIL file on the AutoSys database, type in the following: **jr JOBNAME -q**

To retrieve a copy of the JIL file from the database for updating, type in the following: **jr JOBNAME -q > JOBNAME.xxx.jil** where xxx are your initials.

> *Note: This will create a jil file that will be used to update AutoSys. Follow the published instructions to have this promoted into production (or model).*

When using update, the minimum to code is what needs to be changed. Everything else will remain the same.

```
update_job: PCLD700
date_conditions: 1
days_of_week:mo, tu, we, th, fr
start_times: "17:00"
```

However, any line can be simply modified or changed and updated using the full record.

```
/* ----------------- ELGPD010 ----------------- */


update_job: ELGPD010    job_type: c
box_name: PELDICES
command: autojob.sh ELGJD010
machine: inprodmar
#owner: dsibprod@dsibsun2
permission: mx,me
condition: SUCCESS(ELGPD003) AND SUCCESS(ELGPD005) AND SUCCESS(ELGP
D007) AND NOTRUNNING(ELGPM050) AND SUCCESS(ELGPW010)
description: "ICES Recipient Eligibility Update"
alarm_if_fail: 1
```

Figure 4.4 − Recommended Method for Updating a Jil is to Supply All Information

> *Note: An update_job CANNOT remove a line*
> *previously there, it can only add lines or modify*
> *existing lines. Also, update job CANNOT be used*
> *to change owners. In each case, the job will need*
> *to be deleted and reinserted.*

If a brand new box and job needs to be defined to AutoSys, essentially the same steps should be followed:

- Identify a jil that closely resembles the one needed to be developed. Type the following: **jr JOBNAME -q**

- To retrieve a copy of the JIL file from the database for updating, type in the following: **jr JOBNAME -q > JOBNAME.xxx.jil** where xxx are your initials.

- Use the vi editor in UNIX to modify the file with all of the information needed to meet the requirements.

> *Note: Jobs (or individual boxes) can be deleted by the*
> *following command: **Delete_job: JOBNAME**.*

- However, if an entire box should be deleted including all of the jobs under it, the following command can be used. **Delete_box: BOXNAME**. This option should be used with caution.

> *Note: If a job or box is deleted from AutoSys, it has no*
> *effect on the actual UNIX job. If the intent is to*
> *remove that as well, use the vctl –d option.*

Best Practices

New boxes should be organized so that they flow from the top down. For example, the first job that runs should be listed first with its dependant jobs listed later. Watcher jobs should be listed first. This assists the systems engineer (SE) in predicting which jobs will run next. It also facilitates successing the jobs in reverse order in the case of an abend or missing inputs. For example, if a box of many jobs needs to be successed because of a lack of input, it is important that the SE success the jobs in the reverse order of their dependencies so that marking one job a success does not kick off another job inadvertently. Marking a job SUCCESS that is currently active will not stop the job from running and may give unexpected results.

> *Note: Adding a job to an existing box will ALWAYS
>   show the new job at the bottom of the box
>   regardless of what its true running position
>   might be. For this reason, an SE should never
>   ASSUME that a box is being successed in reverse
>   order just by successing from the bottom of the
>   list.*

If a job depends on the success of a job in another box, the success of the other BOX should be the prerequisite. For example: jobs that wait on the database bounce should wait on **dbap_bounce_inaimp1** not on **dbap_bnce_inaimp1**.

Watcher jobs should be scheduled to run on the day the file is expected, preferably within the hour of data receipt. They can run for several days if they are waiting for a file from an outside vendor such as HMS. The concern here would be if it becomes necessary to bounce AutoSys or sun1. Often that stops a watcher job from functioning even if it still shows RU (running) status. If the job cannot be scheduled very near the expected time of receiving the file, the time interval should be set to 3600 seconds

Section 5: AutoSys Calendars

Custom calendars can be created to either specify dates to run a job
(run_calendar), or dates <u>not</u> to run a job on (exclude_calendar).
Autocal inserts/deletes days in the calendar into the AutoSys database.
Each calendar has a unique name, and a list of days.

```
Autocal_asc <enter>
CALENDAR NAME:                      ORTDAILY    <enter>
Add (a) delete (d) print (p):    a <enter> or    d <enter> or    p <enter>
Date (MM/DD/YY [HH:MM]):            12/10/96 <enter>
```

Figure 5.1 − UNIX Commands to Change/Print Calendar

Instructions on how to add/change an AutoSys calendar can be found at
the following location:
I:SYSTEMS\PSH\PROD_SPT\CYCLEDOC\AUTOSYS\CALENDAR.DOC

The actual excel spreadsheet of calendar dates for the last two years is
in the following file:
I:SYSTEMS\PSH\PROD_SPT\CYCLEDOC\AUTOSYS\AUTOCAL.XLS